Building a Culture of Innovation

By Dr. Peter Denning with John Sanders

It's a common saying that the Department of Defense's greatest challenge is preparing to fight and win the next war with a fighting force that has been exquisitely trained and equipped — to win the last war. Defense leadership knows that the next conflict will include a major cyberspace dimension and has shown that superior battlefield awareness and coordination enables quick victories. To this end, DoD's force transformation objective changes traditional fighting forces to forces consistent with a network-centric philosophy and mode of operations.

Education plays a critical role in force transformation. It is the principal means by which the future warfighter can learn the philosophy and technology of network-centric operations. The Naval Postgraduate School has stepped up to the challenge of preparing future warfighters in numerous ways. The Computer Science Department has taken a leadership role by transforming its curriculum.

In early 2003, the NPS computer science faculty initiated a comprehensive curriculum review. We had two primary objectives: first, to emphasize a principles-oriented approach to computer science and, second, to help students learn to be participants in a culture of innovation as envisioned by the Chief of Naval Operations, Adm. Vern Clark. We tackled the first objective by developing a new framework for studying the great principles underlying all computing technology. We tackled the second objective by designing new courses to help students plan and execute transformative and military-relevant master's theses.

NPS students are professional leaders. Many are Navy commanders and lieutenant commanders, or Army and Marine Corps majors and captains with considerable experience as leaders in their Services. They are highly disciplined, pragmatic and action-oriented. They demand relevance and the simplest and most direct tools to get the job done. Their strong sense of purpose and dedication inspires the faculty to deliver a rigorous and relevant education.

Great Principles of Computing

Our main motivation for developing a principles-based approach is time: Our students have only two years to become competent computing professionals. Although many have backgrounds in computing, it's been five to 10 years since they were in school; the field has changed so much in that time and many are rusty. Some NPS students need to learn computer technology basics.

Computing is about 60-years-old as an academic field of study. The first computer science curricula in the late 1950s had four core courses and a host of technology electives. During the next 40 years, the core curriculum grew slowly and, by 1990, was organized around nine core technologies. Then in the 1990s, with the arrival of the World Wide Web and the dramatic expansion of the Internet, the number of core technologies tripled to about 30. This is far beyond the capacity of a core curriculum. Many universities and their professional societies have been struggling with ways to accommodate this large change in the number of core technologies. We felt the pressure acutely because students must finish their graduate work and thesis research within two years and return to their military duties.

Our new framework has five categories of principles of computing:

√ Computation (models of computers and processing time for computations)
√ Communication (compressing and transmitting data accurately from one site to another)
√ Coordination (the joint actions of human and computer entities to achieve complex common goals)
√ Recollection (naming, storing and retrieving data)
√ Automation (seeking computing alternatives for human cognitive tasks)

Our framework also recognizes three core practices:

√ Design (the layout and construction of computing systems that are dependable, reliable, usable, secure and safe)
√ Development (programming, systems, innovating)
√ Modeling (experiments, data analysis, modeling, prediction, simulation, validation)

The new framework has eight categories rather than 30 in a core technologies approach. It is much easier to grasp and much easier on students. We implemented the framework by creating a new first course, Great Principles of Computing Technology. We reviewed all our other courses so that their syllabi draw on the principles approach and eliminated redundancy. We also separated development (programming and systems) modeling and innovating into a computing practices segment.

We worked with the Operations Research Department to design a modeling practices course, and we created a three-quarter sequence course about innovation. With this framework we have found it is much easier for our students to understand the broad scope of the field and identify the science and engineering principles at work in each computing technology.

Many people are surprised to learn that the computer science faculty completed this change in just six months. The new curriculum was implemented in October 2003. In most public universities, major curriculum revisions take two to three years. The Naval Postgraduate School is quite agile and can change and modernize curricula within months.

I have been the designer and lead instructor in the new Great Principles course. This course has a noble purpose: to introduce the field in terms of its fundamental principles, rather than its core technologies. It serves as a roadmap for the rest of the curriculum for developing strategic, big-picture thinking about our field. The idea of getting directly at the principles of computing is very appealing to our students. For example, during my 35 years as a teacher of computing, Turing Machines (simple abstract computational devices intended to help investigate the extent and limitations of what can be computed) are looked upon as fundamental. Many of our students find them too abstract. So we are finding other ways to explain the limits of computing systems without requiring them to learn Turing Machine theory.

Innovation: A Core Practice of Computing

Let me focus on one other aspect of our new curriculum. In our review, we agreed that innovation is essential for the ongoing creation of wealth and success in businesses and organizations. Yet, most people believe that innovations are often fortuitous occurrences: that it's difficult to predict which ideas will become innovations and how valuable they will be. Therefore, it seems that there is no reliable skill set associated with innovation.

We concluded that these perceptions arise from a general misconception about the nature of innovation, especially the commonly held belief that the work of innovation means the creation of new or novel ideas. The new idea flows through a pipeline of research, development, prototyping, manufacturing and marketing, transforming it into a product or service with an economic impact. Thus, the pipeline is the path for the idea to achieve impact; the inventor is the seed that sets the whole process in motion.

But this model does not explain some of the most successful innovations around us, for example, the Internet and the Linux operating system. Neither of these exemplifies the pipeline model. Linux, for example, has been completely developed, changed and maintained by a large community of volunteers who were not seeking economic gain. Linux didn't begin with a new invention; Unix already existed. It didn't start with a research paper. It started because Linus Torvalds was concerned about making a high-quality, public-domain version of Unix available to the masses. Nobody doubts that Linux was an innovation, and yet it doesn't meet the conventional idea of what an innovation is.

The same thing is true with the World Wide Web. Tim Berners-Lee demonstrated the first browser on a NeXT computer in 1991. He invented it as a proof-of-concept for his idea of document sharing by a worldwide web of interlinked documents. In many ways, the browser was unremarkable because it used many existing technologies. Berners-Lee worked tirelessly to make his technology useful so that people could adopt it into their work. In 1994, he founded the World Wide Web Consortium, W3C, to be a forum where people could reach consensus on Web services and standards to promote the ongoing development of the Web. Berners-Lee never wavered from his conviction that the basic software for the Web should be in the public domain and free to everyone. He repeatedly turned down opportunities to start companies that would allow him to profit from his own invention.

And much the same is true of the Internet. The Internet started as ARPANET, a DoD research project aimed at facilitating resource sharing among DoD computers. During the 1980s, ARPA cooperated with the National Science Foundation, which through a lot of volunteer labor created CSNet and then NSFNET, the backbone of the modern Internet. ARPA also endorsed a consortium, the Internet Society and its Internet Engineering Task Force, which kept the software in the public domain and fostered consensus on protocols and data standards.

The bottom line is that none of these innovations fits the pipeline model. Their technologies were formed from ideas advanced from many directions, but without an identifiable inventor. Most of the work was done by volunteers who had no prospect or interest in economic gain. The real work of innovation is in changing how a community of people thinks and acts — bringing about the adoption of an idea. Although it appears that some people are much better than others at fostering changes in communities, we concluded that we can teach innovation and to do so we must differentiate it from invention.

We define innovation simply as a transformation of practices in a community. We therefore focused on setting up a course that would cultivate the practices an officer needs to effectively produce innovations. We created a three-quarter course: Technology and Transformation. The course has two main objectives: (1) Teaching students how to be self-generating innovators capable of practicing in a culture of innovation, and (2) Helping students plan and execute a transformative master's thesis; the thesis becomes a process of transformation in miniature.

As we gain experience with this framework, we are finding that more and more people are intrigued with the notion that there is a core set of personal practices of innovation. There are many books that tell how an organization can manage itself to be innovative. But there is hardly anything on what the individual must do to be able to participate effectively in a culture of innovation within an organization. This may explain why some guidelines for innovation exist in some companies but not in others: Some groups have the necessary personal practices, others do not.

Innovation as a Skillful Practice

We drew a good deal of inspiration from Peter Drucker, whose 1985 book, *Innovation and Entrepreneurship*, is a gold mine of insights into how innovation really works. It gets to the fundamental issues behind innovation and talks about how individuals and organizations can embrace this process. Drucker defines five phases in the practice of innovation: (1) locate an opportunity; (2) analyze it; (3) assess your community's receptivity; (4) maintain a focus on a simple core idea; and (5) exercise leadership.

The first phase of the innovation process is identifying an opportunity. Drucker lists seven sources of opportunities: (1) the unexpected; (2) incongruities; (3) process needs; (4) change of industry structure; (5) demographics; (6) change of mood or perception; and (7) new knowledge.

The first four show up as challenges to the internal operations of an organization; the other three are external and are subject to competition from other organizations. We added an eighth source to the list, which we call "dead cows." This is a reference to Louis Pasteur, who organized his scientific investigation around the French cattle industry, which was being decimated by anthrax until he invented a vaccine. Major innovations can occur by showing people how to keep their cows healthy.

During phase two, the innovator analyzes the costs, risks, people, strategies and resources needed to effect the change envisioned. In phase three, listening, we meet with members of the target community to assess their degree of receptivity to the proposal and seek their feedback. This phase, which consists of a lot of listening, contrasts with the intellectual bent of the previous phase.

In phase four, focus, we execute the plan devised in phase two and vetted in phase three. This requires constant attention to the simple, central idea behind the mission and a determination to avoid being sidetracked by interesting — but nonessential ideas and opportunities. The final phase, leadership, is a commitment to excellence in product and service, a commitment to do the work needed to win acceptance of the proposal.

It's not hard to identify the skills needed to accomplish these phases: awareness, focus, persistence, listening, blending and simplicity. In addition, you need skills for: making powerful declarations and compelling offers; leading a team that will help you carry out the plan; being a constant learner; and maintaining a sense of destiny. You also need a sense that you are acting on behalf of a purpose larger than yourself.

Innovation in Network-Centric Operations

Let me give you an example that has led to a project that may produce an innovation of great value to the Navy and DoD. I discussed earlier the DoD's interest in adapting its warfighting doctrine to a highly networked world. DoD leadership has laid out plans to develop a Global Information Grid (GIG), a worldwide network capable of supporting future military operations. In this setting, military operations are called network-centric operations (NCO).

Leadership has been increasingly frustrated at what they see as painfully slow progress toward implementing the GIG. Interestingly, network engineers have also experienced frustration. They see a large number of guidance documents coming from DoD, Navy, Army and Air Force, but there is no authority that can resolve important but relatively low-level engineering ambiguities and conflicts. Everyone is frustrated — leadership because the engineers are not moving fast enough — and the engineers because leadership has not provided a method to resolve ambiguities and conflicts.

In response to this need, we proposed a new entity: W2COG or World Wide Consortium for the Grid. The W2COG is a consortium of government, industry and academic engineers working on the continuing goal of advancing networking technology to support the GIG. The W2COG aims to accelerate systems interoperability agreements between units and agencies in a highly complex environment where technical guidance can never be complete, there is no central authority and both the technology and environment are constantly changing. Strengthening GIG technology will enable more robust joint warfighting capabilities.

This consortium is modeled after the highly successful W3C. Thus, the W2COG will provide an agile, fast-response consensus process that enables the members to reach agreements on data formats, protocols, information exchange patterns, and other aspects of interoperability that are needed to enable systems connected to the GIG.

W2COG will produce recommendations, guidelines, models and tools. It will deal only with open architectures, recommendations and consensus processes, but it will not produce standards because there are other organizations tasked with that purpose. W2COG will be hosted by the NPS, just as the Massachusetts Institute of Technology (MIT) hosts the W3C.

To achieve these goals, we are working toward a strategic partnership with NCOIC, the Network Centric Operations Industry Consortium. The two consortia would share reciprocal membership rights and jointly operate the technical agenda. This partnership would create a single umbrella for government, industry and academia to work together to advance the technology for NCO.

The objective — networking to support NCO — is a moving target because it depends on military strategy, defense doctrine and information technology — all of which are constantly changing and reshaping. Current acquisition, planning and technology development systems move too slowly to enable us to close the distance to the target. Moreover, the complexity of the networking technology and inter-organizational coordination is beyond the scope of any one authority. The consortium model is the only realistic alternative with a prospect of reaching the goal.

Institutionalizing Innovation

We have learned a great deal since October 2003 when we began our new curriculum. We recently formed a group composed of Association for Computing Machinery (ACM) award winners to create a Great Principles Framework that might extend to computer science education at other universities.

We recognize that our students and alumni must become self-generating innovators. They must be leading practitioners who can continuously leverage knowledge superiority in the Navy's culture of innovation. NPS computer science graduates will be agents of change who will help the United States maintain a technological and operational advantage.

Dr. Peter Denning is chairman of the Naval Postgraduate School Computer Science Department and director of the NPS Cebrowski Institute for Information Innovation and Superiority. He is one of the founders of CSNet.

John Sanders, NPS Director of University Relations, contributed to this article.

CHIPS